DOCUMENT RESUME

ED 051 268 TM 000 579

AUTHOR Jensema, Carl; Urry, Vern W.
TITLE A Program for Oblique and Orthogonal Procrustes Rotation.
INSTITUTION Washington Univ., Seattle. Bureau of Testing.
PUB DATE Jan 71
NOTE 12p.

EDRS PRICE EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS Algorithms, Computer Programs, Factor Analysis, Factor Structure, *Goodness of Fit, Mathematics, *Oblique Rotation, *Orthogonal Rotation, *Transformations (Mathematics)

ABSTRACT

Procrustes rotation involves fitting a factor pattern matrix to a specified target matrix in factor analysis. These rotations are useful for the investigator who wishes to see how well his data can be made to fit a hypothesized factor pattern matrix. The mathematical problems involved in these transformations are outlined and computer algorithms for both oblique and orthogonal solutions are presented. (CK)

ED051268

A Program for Oblique and Orthogonal

Procrustes Rotation

Carl Jensema

and

Vern W. Urry

Bureau of Testing
University of Washington—Seattle

TM 000 579

Bureau of Testing

University of Washington

January 1971

A Program for Oblique and Orthogonal Procrustes Rotation

Carl Jensema

and

Vern W. Urry

Abstract

Procrustes rotation involves fitting a factor pattern matrix to a specified target matrix. This report briefly outlines the mathematical problems involved and provides computer algorithms for both oblique and orthogonal solutions.

Bureau of Testing Project: 0171-178

Bureau of Testing

University of Washington

January 1971

A Program for Oblique and Orthogonal Procrustes Rotation

Carl Jensema

and

Vern W. Urry

Those familiar with Greek mythology will remember that Procrustes was a character whom Theseus encountered in his extended travels. Procrustes took pride in having beds which fitted all travelers: those persons who were too tall he cut down to size and those who were too short he stretched. The name, Procrustes, is particularly appropriate for the factorial rotation methods to be discussed here.

It sometimes happens in the course of research involving factor analysis that an investigator may want to see how well his data can be made to fit a hypothesized factor pattern matrix. The problem may be stated more formally as: Given a factor pattern matrix, A, and a hypothesized factor pattern matrix, B, how may a transformation matrix, T, be obtained such that

$$AT = B + E$$

is satisfied with the restrictions that

$$tr\ (E'E) = \text{minimum}$$

and

$$D_{T'T} = I \quad ?$$

Horst (1956) seems to have been one of the first to propose a workable solution for this. Ahmavaara (1957) considered the problem, as did Hurley and Cattell (1962). All of these authors gave essentially the same oblique solution.

The solution is very easily computed. Solving

$$T^* = (A'A)^{-1}A'B$$

and normalizing the columns of T^* gives T, the desired transformation matrix. Multiplying to obtain AT gives a matrix which is a least squares fit of B with the restriction that

$$D_{T'T} = I.$$

The above solution is oblique. An orthogonal solution utilizes the additional constraint that

$$T'T = TT' = I.$$

This somewhat more difficult problem was examined in detail by Schönemann (1966). His method begins by taking the minor product of A and B as

$$S = A'B.$$

then calculating $S'S$ and SS'. Obtaining all roots and vectors of these two matrices, preferably by a Jacobi method, gives

$$S'S = VDV'$$

and

$$SS' = WDW'.$$

A problem occurs at this stage concerning the proper orientation of the orthogonal vectors in V and W. This is quite easily solved by computing

$$W'SV = D^{\frac{1}{2}}$$

and checking each diagonal element in $D^{\frac{1}{2}}$. If the diagonal element is negative, the corresponding column of W is reflected.

With the new reflected matrix, W*, we may proceed to obtain

$$T = W^*V'$$

as the desired orthogonal transformation matrix. Calculating AT gives an orthogonal least squares fit of B.

References

Ahmavaara, Y. On the unified factor theory of mind. Ann. Acad. Sci. Fenn., Ser. B. 106, Helsinki, Finland, 1957.

Horst, P. A simple method for rotating a centroid factor matrix to a simple structure hypothesis. Journal of Experimental Education, 1956, 24, 251-258.

Hurley, J. R., and Cattell, R. B. The procrustes program: producing direct rotation to test a hypothesized factor structure. Computers in Behavioral Science, 1962, 258-262.

Schönemann, P. H. A generalized solution to the orthogonal procrustes problem. Psychometrika, 1966, 31, 1-10.

"PROCRUS"

Program Setup Instructions

A. Heading Card -- Any alphanumeric message to be printed at beginning of output.

B. Problem Card

Column	Item
1 - 7	Punch "PROBLEM".
8 -12	Iteration tolerance for Jacsim Root and vector Subroutine. (Try 00100 or 00010.)
13-14	Number of variables in each matrix
15-16	Number of factors in each matrix (Note--the loading matrix and the target matrix must be the same size)

C. Input Variable Format Card. -- One card with any F-type format. Both input matrices must have the same format.

D. Date--Must be on cards. Matrices must have variables in rows, factors in columns. Input two complete matrices in the following order.

1. Factor loading matrix to be transformed.
2. Target matrix.

This is a repeating program. Any number of problems may be run by repeating steps A through D.

E. Finish -- At the end of all problems place a card with "FINISH" punched in columns 1-6.

```
RUNX(S).
LGO.
-00
      PROGRAM PROCRUS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION A(25,10),B(25,10),AT(25,10),T(10,10)
     *,HDG(8),FMT(8)
C     THIS PROGRAM WAS WRITTEN BY CARL JENSEMA IN JANUARY, 1971.
C     IT PROVIDES BOTH OBLIQUE AND ORTHOGONAL PROCRUSTES SOLUTIONS.
C     THE METHODS COME FROM PSYCHOMETRIKA (MARCH,1966,P.1-10).
C     A= INPUT LOADING MATRIX.
C     B= TARGET MATRIX.
C     T= TRANSFORMATION MATRIX.
C     AT= TRANSFORMED LOADINGS.
C     TOL= ITERATION TOLERANCE.
C     NV= NUMBER OF VARIABLES.
C     NF= NUMBER OF FACTORS.
10    READ(5,1)HDG,PROB,TOL,NV,NF,FMT
1     FORMAT(8A10/A7,F5.5,2I2/8A10)
      IF(PROB.NE.7HPROBLEM) GO TO 10000
      IF(HDG(1).EQ. 10HFINISH     ) GO TO 10000
      IF(PROB.EQ. 7HFINISH ) GO TO 10000
      IF(FMT(1).EQ.10HFINISH     ) GO TO 10000
      WRITE(6,2) HDG,TOL,NV,NF,FMT
2     FORMAT(45H1UNIVERSITY OF WASHINGTON BUREAU OF TESTING
     1//34H PROCRUSTES TRANSFORMATION PROGRAM//1X,8A10
     2///20H ITERATION TOLERANCE                 10X,F9.5
     3// 20H NUMBER OF VARIABLES                 10X,I3
     4//20H NUMBER OF FACTORS                    10X,I3
     5///15H INPUT FORMAT              8A10)
      WRITE(6,3)
3     FORMAT(5(/),15H INPUT LOADINGS )
      DO 20 I=1,NV
      READ(5,FMT) (A(I,J),J=1,NF)
      WRITE(6,4) I,(A(I,J),J=1,NF)
4     FORMAT(// 4H ROW   I3,(/15F9.3))
20    CONTINUE
      WRITE(6,5)
5     FORMAT(5(/),14H TARGET MATRIX)
      DO 30 I=1,NV
      READ(5,FMT)(B(I,J),J=1,NF)
      WRITE(6,4)I,(B(I,J),J=1,NF)
30    CONTINUE
      DO 50 I=1,2
      IF(I.EQ.1) CALL OBLIPRO(A,B,AT,T,TOL,NV,NF)
      IF(I.EQ.2) CALL ORTHPRO(A,B,AT,T,TOL,NV,NF)
      WRITE(6,6)
6     FORMAT(5(/),22H TRANSFORMATION MATRIX)
      DO 40 J=1,NF
      WRITE(6,4)J,(T(J,K),K=1,NF)
40    CONTINUE
      WRITE(6,7)
7     FORMAT(5(/),21H TRANSFORMED LOADINGS )
      DO 50 J=1,NV
      WRITE(6,4)J,(AT(J,K),K=1,NF)
50    CONTINUE
      GO TO 10
```

```
10000 WRITE(6,8)
8      FORMAT(10(/),12H END OF JOB.)
       STOP
       END




       SUBROUTINE OBLIPRO(A,B,AT,T,TOL,NV,NF)
C      OBLIQUE PROCRUSTES
       DIMENSION A(25,10),B(25,10),AA(20,10),AB(10,10),T(10,10),
      *AT(25,10),E(10)
       WRITE(6,1)
1      FORMAT(10(/),1X,21(1H*)//20H OBLIQUE SOLUTION    //1X,21(1H*))
C      OBTAIN MINOR PRODUCT MOMENT OF LOADING MATRIX.
       DO 10 I=1,NF
       DO 10 J=1,NF
       AA(I,J)=0.0
       DO 10 K=1,NV
       AA(I,J)=AA(I,J)+A(K,I)*A(K,J)
10     CONTINUE
C      OBTAIN ROOTS AND VECTORS.
       CALL JACSIM(AA,E,TOL,NF)
C      ROOTS ARE IN DIAGONAL OF UPPER PART OF AA AND ALSO IN E.
C      EIGNVECTORS ARE COLUMNS IN LOWER PART OF AA.
C      INVERT MATRIX.
       DO 20 I=1,NF
       E(I)=1.0/E(I)
20     CONTINUE
       DO 30 I=1,NF
       INF=I+NF
       DO 30 J=1,NF
       JNF=J+NF
       AA(I,J)=0.0
       DO 30 K=1,NF
       AA(I,J)=AA(I,J)+AA(INF,K)*E(K)*AA(JNF,K)
30     CONTINUE
C      MULTIPLY THE LOADING MATRIX BY THE TARGET MATRIX
       DO 40 I=1,NF
       DO 40 J=1,NF
       AB(I,J)=0.0
       DO 40 K=1,NV
       AB(I,J)=AB(I,J)+A(K,I)*B(K,J)
40     CONTINUE
C      MULTIPLY AA BY AB TO GET TRANSFORM MATRIX.
       DO 50 I=1,NF
       DO 50 J=1,NF
       T(I,J)=0.0
       DO 50 K=1,NF
       T(I,J)=T(I,J)+AA(I,K)*AB(K,J)
50     CONTINUE
C      NORMALIZE THE TRANSFORMATION MATRIX BY COLUMNS
       DO 70 J=1,NF
       SUMSQ=0.0
       DO 60 I=1,NF
```

```
      SUMSQ=SUMSQ+(T(I,J)**2)
60    CONTINUE
      SUMSQ=SQRT(SUMSQ)
      DO 70 I=1,NF
      T(I,J)=T(I,J)/SUMSQ
70    CONTINUE
C     MULTIPLY LOADINGS BY TRANSFORM
      DO 80 I=1,NV
      DO 80 J=1,NF
      AT(I,J)=0.0
      DO 80 K=1,NF
      AT(I,J)=AT(I,J)+A(I,K)*T(K,J)
80    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ORTHPRO(A,B,AT,T,TOL,NV,NF)
C     ORTHOGONAL PROCRUSTES, SCHONEMANN METHOD(PSYCHOMETRIKA,MARCH,1966)
      DIMENSION A(25,10),B(25,10),BAAB(20,10),ABBA(20,10),
     *AB(10,10),T(10,10),AT(25,10),E(10)
      WRITE(6,1)
1     FORMAT(10(/),1X,21(1H*)//20H ORTHOGONAL SOLUTION //1X,21(1H*))
C     OBTAIN MINOR PRODUCT OF LOADINGS AND TARGET.
      DO 10 I=1,NF
      DO 10 J=1,NF
      AB(I,J)=0.0
      DO 10 K=1,NV
      AB(I,J)=AB(I,J)+A(K,I)*B(K,J)
10    CONTINUE
C     OBTAIN MINOR AND MAJOR PRODUCTS OF AB.
      DO 20 I=1,NF
      DO 20 J=1,NF
      BAAB(I,J)=0.0
      ABBA(I,J)=0.0
      DO 20 K=1,NF
      BAAB(I,J)=BAAB(I,J)+AB(K,I)*AB(K,J)
      ABBA(I,J)=ABBA(I,J)+AB(I,K)*AB(J,K)
20    CONTINUE
C     OBTAIN ROOTS AND VECTORS
      CALL JACSIM(BAAB,E,TOL,NF)
      CALL JACSIM(ABBA,E,TOL,NF)
C     VECTORS ARE IN LOWER PART OF THE MATRICES.
C     BEGIN REFLECTION OF ABBA VECTORS.
C     (SEE P.8, BOTTOM PARAGRAPH OF SCHONEMANN ARTICLE)
C     MULTIPLY TO OBTAIN WSV.
      DO 30 I=1,NF
      DO 30 J=1,NF
      ABBA(I,J)=0.0
      DO 30 K=1,NF
      KNF=K+NF
      ABBA(I,J)=ABBA(I,J)+ABBA(KNF,I)*AB(K,J)
30    CONTINUE
```

```
      DO 40 I=1,NF
      DO 40 J=1,NF
      AB(I,J)=0.0
      DO 40 K=1,NF
      KNF=K+NF
      AB(I,J)=AB(I,J)+ABBA(I,K)*BAAB(KNF,J)
40    CONTINUE
C     CHECK DIAGONAL OF WSV, REFLECT COLUMN IF DIAGONAL ELEMENT IS NEGATIVE.
      DO 50 I=1,NF
      IF(AB(I,I).GE.0.) GO TO 50
      DO 50 J=1,NF
      JNF=J+NF
      ABBA(JNF,I)=ABBA(JNF,I)*(-1.)
50    CONTINUE
C     END OF REFLECTION
C     OBTAIN TRANSFORMATION MATRIX BY MULTIPLYING VECTORS.
      DO 60 I=1,NF
      INF=I+NF
      DO 60 J=1,NF
      JNF=J+NF
      T(I,J)=0.0
      DO 60 K=1,NF
      T(I,J)=T(I,J)+ABBA(INF,K)*BAAB(JNF,K)
60    CONTINUE
C     MULTIPLY LOADINGS BY TRANSFORM.
      DO 70 I=1,NV
      DO 70 J=1,NF
      AT(I,J)=0.0
      DO 70 K=1,NF
      AT(I,J)=AT(I,J)+A(I,K)*T(K,J)
70    CONTINUE
      RETURN
      END
      SUBROUTINE JACSIM(R,D,P,N)
C     THIS SUBROUTINE CALCULATES ALL ROOTS AND VECTORS OF A MATRIX
C     USING A JACOBI METHOD.
      DIMENSION R(20,10),D(10)
      N1=N+1
      N11=N-1
      N2=N*2
      DO 10 I=N1,N2
      DO 10 J=1,N
      R(I,J)=0.0
10    CONTINUE
      DO 20 I=1,N
      NI=N+I
      R(NI,I)=1.0
20    CONTINUE
      DO 90 L=1,100
      DO 30 I=1,N
      D(I)=R(I,I)
30    CONTINUE
      DO 50 I=1,N11
      I1=I+1
      DO 50 J=I1,N
      DR=R(I,I)-R(J,J)
```

```
      A=SQRT(DR**2+4.*R(I,J)**2)
      A=SQRT((A+DR)/(2.*A))
      B=SQRT(1.-A**2)
      C=SIGN(1.,R(I,J))
      DO 40 K=1,N2
      U=R(K,I)*A*C+R(K,J)*B
      R(K,J)=-R(K,I)*B*C+R(K,J)*A
      R(K,I)=U
40    CONTINUE
      DO 50 K=1,N
      U=R(I,K)*A*C+R(J,K)*B
      R(J,K)=-R(I,K)*B*C+R(J,K)*A
      R(I,K)=U
50    CONTINUE
      DO 60 I=1,N
      D(I)=ABS(D(I)-R(I,I))
60    CONTINUE
      S=0.0
      DO 70 I=1,N
      S=MAX1(S,D(I))
70    CONTINUE
      DO 80 I=1,N
      D(I)=R(I,I)
80    CONTINUE
      IF(S-P) 100,100,90
90    CONTINUE
100   RETURN
      END
```